LA-UR-18-24236
Approved for public release; distribution is unlimited.

Title: Weight Window Based Variance Reduction Introduction & Overview

Author(s): Spencer, Joshua Bradly
Martz, Roger Lee
Alwin, Jennifer Louise

Intended for: Attila4MC Class

Issued: 2018-05-15

Weight Window Based Variance Reduction

Introduction & Overview

Joshua B. Spencer, Roger L. Martz, Jennifer L. Alwin
XCP-3: Monte Carlo Codes, Methods and Applications

MCNP® Trademark

MCNP® and Monte Carlo N-Particle® are registered trademarks owned by Los Alamos National Security, LLC, manager and operator of Los Alamos National Laboratory. Any third party use of such registered marks should be properly attributed to Los Alamos National Security, LLC, including the use of the ® designation as appropriate.

- **Please note that trademarks are adjectives and should not be pluralized or used as a noun or a verb in any context for any reason.**
- **Any questions regarding licensing, proper use, and/or proper attribution of Los Alamos National Security, LLC marks should be directed to trademarks@lanl.gov.**

This Section's Objective

- **Introduce the user to the Weight Windows (WW) Variance Reduction Technique (VRT) and its implementation in MCNP**
- **Introduction to Consistent Adjoint Driven Importance Sampling (CADIS) and Forward Weighted (FW) CADIS extension to WW VRT**

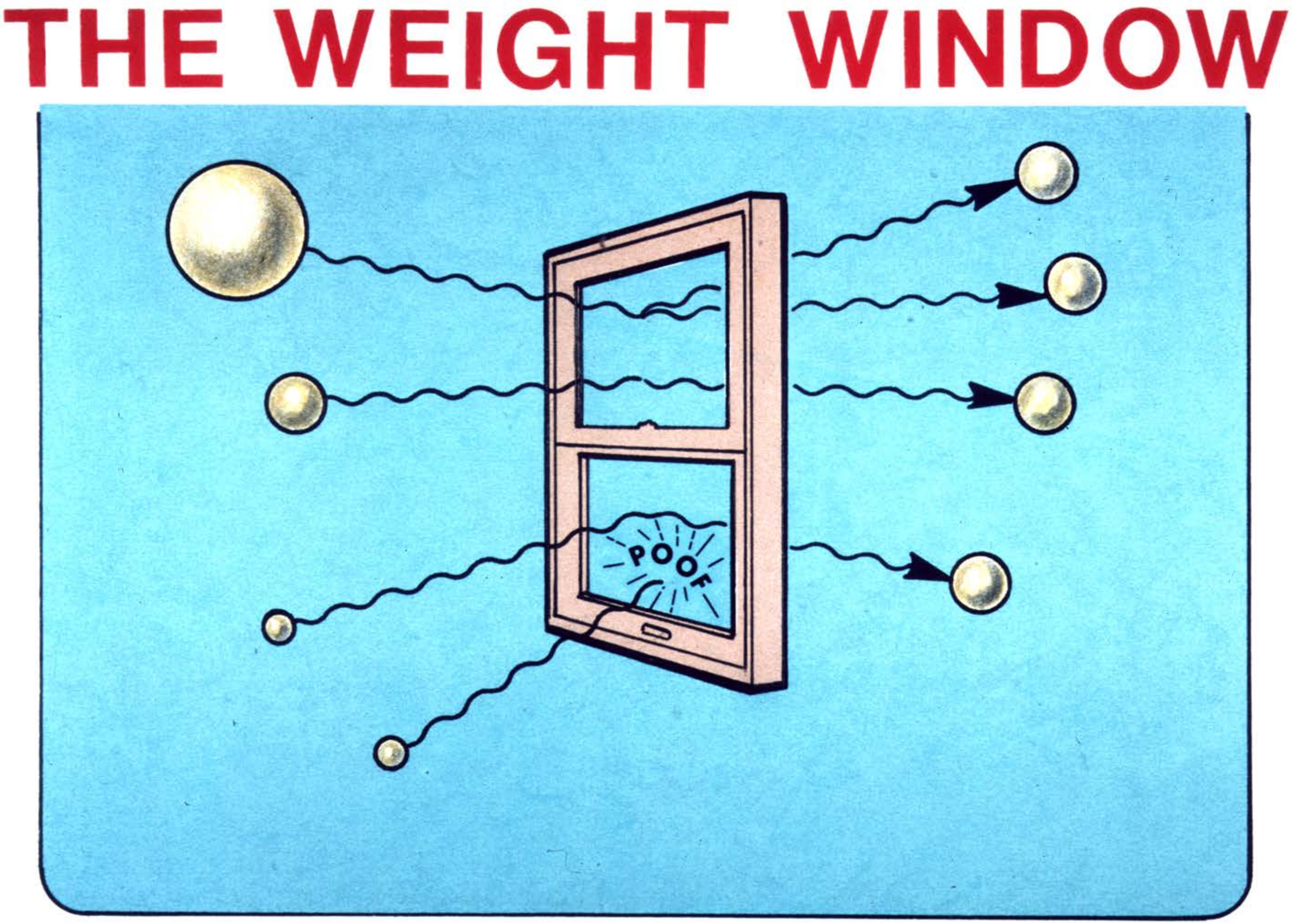
A Variance Reduction Golden Rule

In fixed-source, deep-penetration problems (i.e., not kcode),

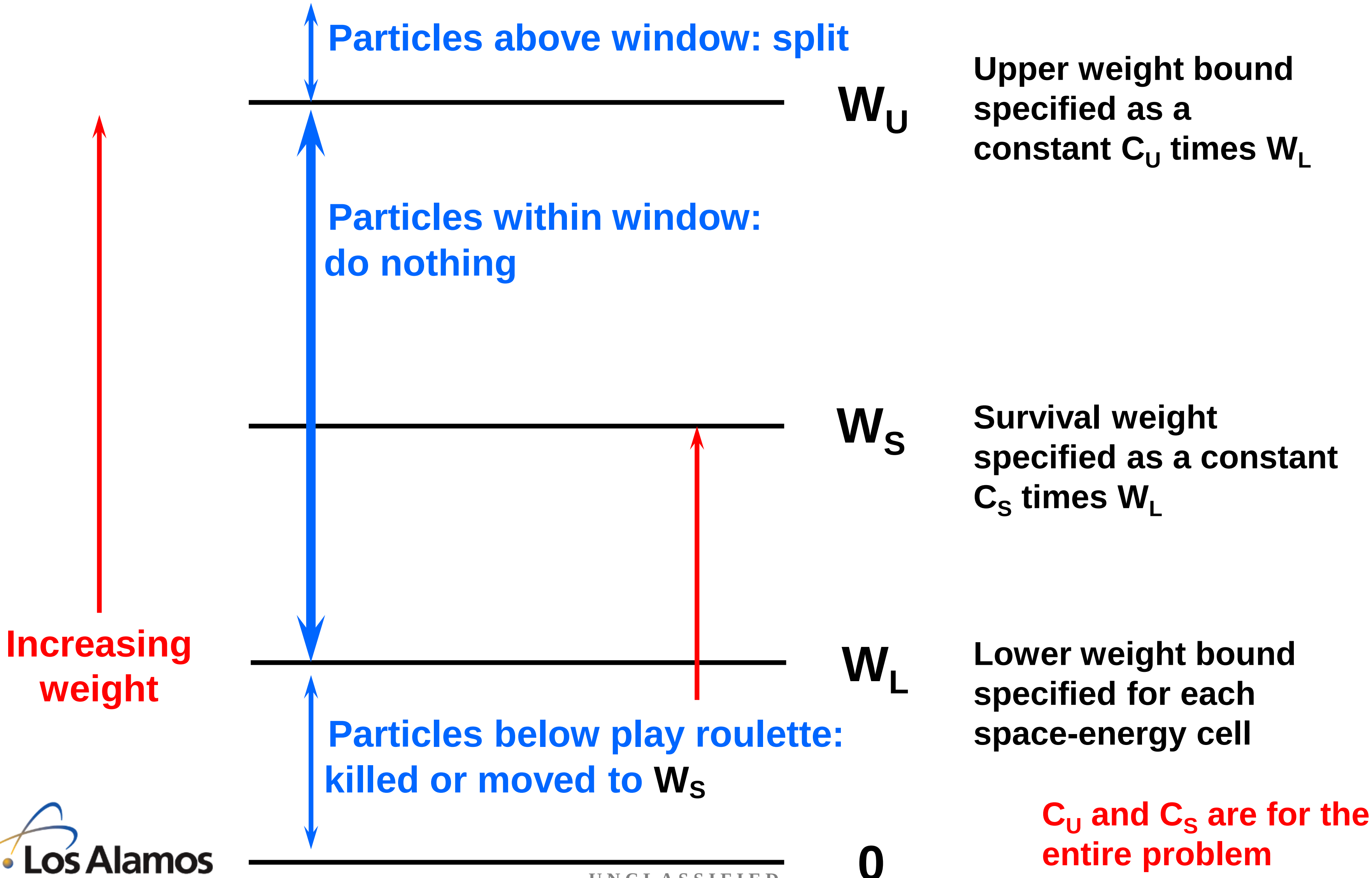
A "few" large weight particles usually are responsible for most of the variance.

If you understand why they exist, then you "might" be able to determine a way to improve the calculation.

Objective: Keep particle weight within specified bounds.



Variance Reduction Techniques



Weight Windows

This is a phase-space splitting and Russian roulette technique.

The phase-space may be:

- **Space**
- **Space-energy**
- **Space-time**
- **Space-energy-time**

Weight Windows

- **Increase transport efficiency by equalizing tally weights.**
  - If every history contributes the same score, a zero-variance solution results.
  - Controls weight fluctuations produced by other VRT's.
- **Inversely proportional to the importance**
  - High importance region has small weight windows
  - Zero window means use weight cutoff
  - -1 window means zero importance, particle terminated
- **Splitting / roulette can be limited to n for 1**
- **Weight control applied at collisions and/or surfaces for cell-based weight windows (plus every 1 m.f.p. for mesh-based windows)**
- **Weight windows turns off geometry splitting & geometry roulette**

Weight Windows

Although the weight window can be effective when used alone, it was designed for use with other biasing techniques that introduce a large variation in particle weight.

Weight Windows

- **Using a window inversely proportional to the importance can ensure that the mean score from any track in the problem is roughly constant.**
  - "*Ideally*" the window should be set such that the track weight times the mean score is approximately constant, enabling identical tally contributions (thus minimizing the variance).
- **Under these conditions, the variance is mostly attributed to the variation in the number of contributing tracks rather than the variation in the track score.**

Weight Windows Implementation in MCNP

- **Cell Based Weight Windows**
- **Mesh Based Weight Windows**

Weight Windows Parameter Card – WWP:n

Weight Windows Parameter Card

`WWP:<pl> WUPN WSURVN MXSPLN MWHERE SWITCHN MTIME`

`<pl>` = **N (neutrons), P (photons), E (electrons)**

`WUPN` = **Sets upper window bound as a multiple of lower window bound.**

Required: `WUPN` ≥ `2`

`WSURVN` = **Sets weight of particle surviving the Russian roulette game as a multiple of the lower weight bound.**

Required: `1 < WSURVN < WUPN`

`MXSPLN` = **Sets maximum splitting/roulette factor**

Required: `MXSPLN > 1`

`MWHERE` = **Sets where to check a particle's weight.**

-1/0/1 = at collisions only / surfaces <u>and</u> collisions / surfaces only
(a value of 1 does NOT turn off the 1 mfp checking for mesh windows)

Weight Windows Parameter Card (cont.)

SWITCHN	= tells MCNP where to get the lower weight window bounds
-1/0	= from an external WWINP file / WWNi cards
>0	= SWITCHN divided by cell importances from IMP card
MTIME	= 0 energy-dependent windows (WWE card)
	= 1 time-dependent windows (WWE card)

Defaults:

```
WUPN    = 5               MWHERE  = 0
WSURVN  = 0.6 * WUPN      SWITCHN = 0
MXSPLN  = 5               MTIME   = 0
```

EXAMPLE: `WWP:N 4 3 3 -1 0 1`

- set upper bound at 4 times lower bound
- set survival weight at 3 times lower bound
- limit splitting/roulette to 3
- check weight at collisions only
- get lower bounds from WWNi cards
- run space-time-dependent windows

Weight Window Input Cards

For cell-based weight windows:

- **Weight window (lower) bounds**

WWNi:n w_{i1} w_{i2} … W_{ij} (j = number of cells)

W_{ij} = lower weight bound in cell j and energy interval E_{i-1}

to E_i as defined on the WWE card

- **Weight window energies (or times)**

WWE:n E_1 E_2 … E_j (E_j <= 99)

WWT:n T_1 T_2 … T_j (T_j <= 99)

E_j = upper energy (or time) bound of the i^{th} window group
If this card is omitted, a single energy (or time) group is assumed.

Mesh-Based Weight Windows

- It is no longer necessary to divide geometry solely to use weight windows.
- The user can supply a superimposed mesh for the generator.
- Simplified geometry setup, significantly saving user time.

Mesh-Based Weight Windows

Some Cautions:

- **Mesh has its own coordinates. Be careful not to confuse particle coordinates with mesh coordinates.**
- **The superimposed mesh should full cover the geometry.**
- **A line or surface should not be made coincident with a mesh surface.**
- **Avoid putting source exactly on mesh boundaries.**
- **Do not use too many or too few mesh cells.**
- **Generator is statistical.**

Mesh Card (1)

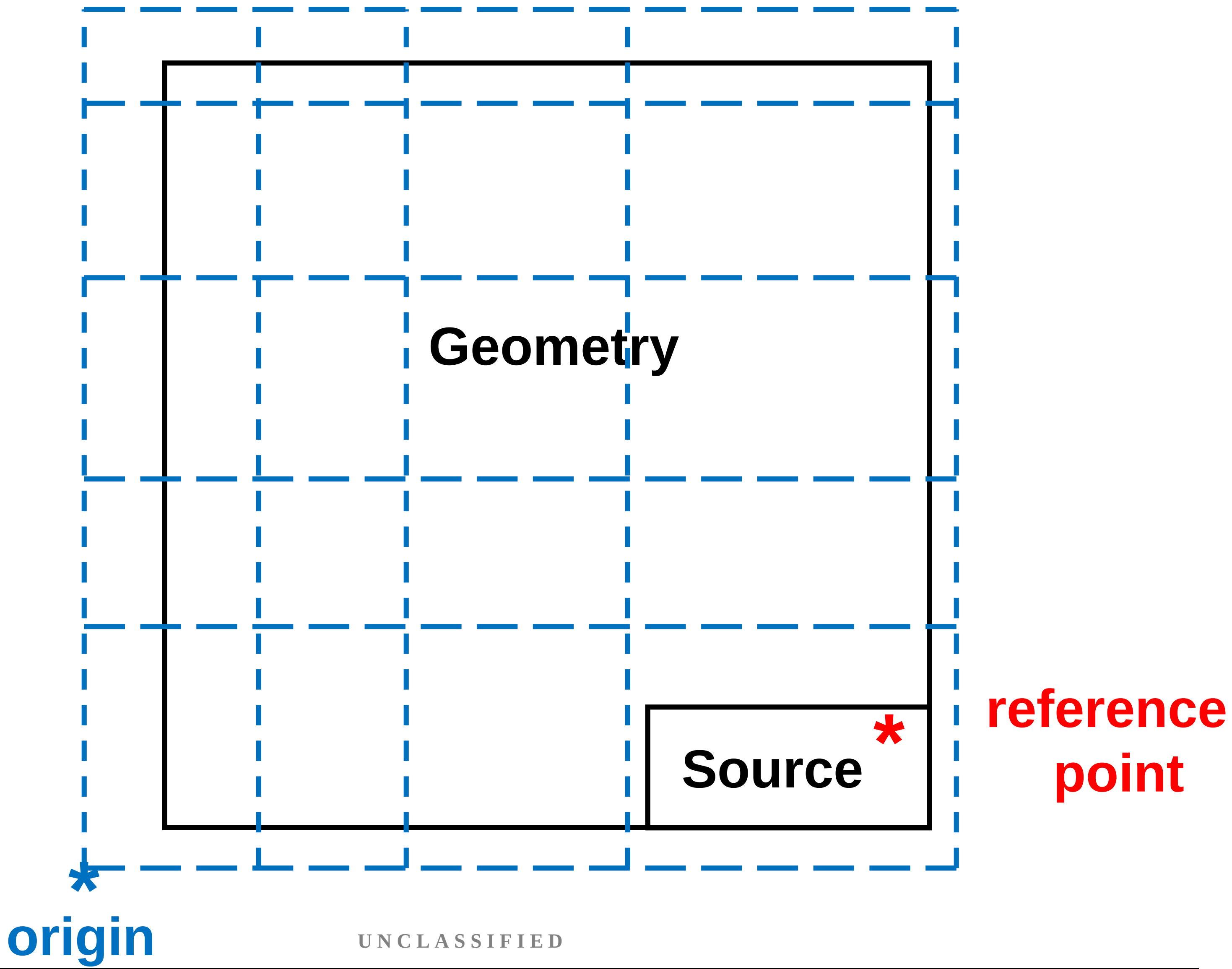
FORM: MESH <mesh variable> = <specification>

Variable	Default	Meaning
GEOM	xyz	Mesh geometry – Cartesian (“xyz” or “rec”) or cylindrical (“rzt” or “cyl”).
REF	none	x, y, z coordinates of the reference point.
ORIGIN	0., 0., 0.	x, y, z coordinates in MCNP cell geometry of the superimposed mesh’s origin.
AXS	0., 0., 1.	Vector giving the direction of the axis of the cylindrical or spherical mesh.
VEC	1., 0., 0.	Vector defining, along with AXS, the plane for $\theta = 0$.
IMESH	none	Locations of coarse meshes in the x direction for rectangular geometry or in the r direction for cylindrical geometry.

Mesh Card (2)

Variable	Default	Meaning
IINTS	**1**	**# of fine meshes in corresponding coarse meshes in x and r (for rec and cyl geometry, respectively).**
JMESH	**none**	**Locations of coarse meshes in the y direction for rectangular geometry or in the z direction for cylindrical geometry.**
JINTS	**1**	**# of fine meshes in corresponding coarse meshes in y and z (for rec and cyl geometry, respectively).**
KMESH	**none**	**Locations of coarse meshes in the z direction for rectangular geometry or in the θ direction for cylindrical geometry.**
KINTS	**1**	**# of fine meshes in corresponding coarse meshes in z and θ (for rec and cyl geometry, respectively).**

Weight Windows Mesh Schematic



Hints

1. **Enter only upper mesh bounds**
2. **Cylindrical mesh origin should not be on axis**
3. **Cylindrical mesh: $0 < \theta < 1$**
   Note: Angular units are “revolutions”. One revolution is 360 degrees.
4. **θ and φ can be in revolutions, radians, or degrees:**
   $(0<\theta\leq 1)$, $(0<\theta\leq 2\pi)$, $(0<\theta\leq 360)$
   $(0<\varphi\leq 0.5)$, $(0<\varphi\leq \pi)$, $(0<\varphi\leq 180)$

Cylindrical Mesh Example

```
mesh   geom=rzt  ref=0 0.001 0
       origin=0.001 -0.001  0
       axs=0 1 0   vec=1 0 0
       imesh 50  101   iints 1 1
       jmesh 50  101   jints 5 5
       kmesh  1        kints 1
```

Rectangular Mesh Example

```
mesh   geom=rec  ref=0  0.001  0
       origin= -150. -0.001  -150.
       imesh 150  250     iints 1 1
       jmesh  50  101     jints 5 5
       kmesh 150  250     kints 1 1
```

MESH Card (6)

Spherical Mesh Example

```
mesh   geom=sph   ref=0   0.001   0
       origin=0  0  0
        axs=0 1 0    vec=1 0 0
       imesh 50  101      iints 20 20
       jmesh 180          jints 4
       kmesh 360          kints 4
```

Setting Weight Window Parameters

- **MCNP Weight Window Generator**
  - Stochastic Tool to Estimate Weight Window Parameters
  - Iterative Solution
- **Use A Deterministic Solution of the Adjoint**
  - Importance is proportional to the adjoint solution if the adjoint source is taken to be the response function for the tally that is being optimized
  - Attila, Varex Imaging
- **Consistent Adjoint Driven Importance Sampling (CADIS)**
  - Weight windows with source biasing → provide particles with weights inside the weight window bounds i.e. biased source weights consistent with weight window grid
  - Attila4MC, Varex Imaging → Supports MCNP6.2
  - Advantage 3.0.3/Denovo, Oak Ridge National Lab → Supports MCNP5 v 1.6
- **Forward Weighted CADIS**
  - Global Variance Reduction by defining a global adjoint source $q = \frac{R(r,E,\Omega)}{\iint R(r,E',\Omega')\psi(r,E',\Omega')dE'd\Omega}$

    R is a desired response function and ψ is the forward flux calculated deterministically. Once the adjoint source is calculated apply the normal CADIS VRT.
  - Attila4MC v10.0 coming soon, Varex Imaging → Supports MCNP6.2
  - Advantage 3.0.3/Denovo, Oak Ridge National Lab → Supports MCNP5 v 1.6

WWG Exercises

Weight Window Generator

The weight window generator is a statistical tool built into MCNP for generating lower window values for the weight window.

The window generator has proved very useful; two caveats are appropriate:

- The generator is by no means a panacea for all importance sampling problems.
- It is not a substitute for thinking on the user's part.

Weight Window Generator

$$ES = \frac{TS}{TW}$$

- **ES = Expected score (importance)**
- **TS = Total Score attributed to particles and their progeny entering the cell**
- **TW = Total Weight entering the cell (re-entrant weight not counted)**

MCNP assigns weight windows inversely proportional to the importances.

Weight Window Generator Limitations

- **The principal problem encountered when using the generator is bad estimates of the importance function because of the statistical nature of the generator.**
  - Several iterations (~ 3) could be required before the optimum importance function is found for a given tally.
  - Number of iterations is a function of the geometric complexity.
- **The WWG will also fail when phase-space is not sufficiently subdivided and no single weight window bounds is representative of the whole region. (Use superimposed mesh grid.)**
- **The WWG will also fail if the phase-space is too finely subdivided and subdivisions are not adequately sampled.**

Weight Window Generator Input Card

WWG I_t I_c WG J J J J IE

- I_t = problem tally number (n of the Fn card)
- I_c = invokes cell- or mesh-based weight window generator
  - \> 0 means cell-based with I_c as reference cell (typically a source cell)
  - 0 means use mesh-based generator (MESH) card
- WG = value of generated lower WW bound for cell I_c or for reference mesh (MESH card)
  - 0 → lower bound is one-half the average source weight
- J = unused
- IE = toggles energy- or time-dependent weight windows
  - 0 = WWGE card means energy bins
  - 1 = WWGE card means time bins

Default: no WW values generated. Use is optional.

Weight Window Generator Energies & Time

`WWGE:<pl>` E_1 E_2 … E_n

`WWGT:<pl>` T_1 T_2 … T_n

`E,T` = upper energy, time bound for the i'th weight window energy, time group

Default: one energy or time bin **Use: optional**

Note: If WWGE and/or WWGT specified, then the WWOUT file will have the space/time/energy windows and the WWONE file will have space only windows.

Variance Reduction Exercise #4A

Generate weight windows on a cylindrical mesh while using cell-based weight windows.

- **Copy your input file from Exercise #3B.**
- **Change the `wwg` card to:**

```
wwg  1  0
```

- **Keep the wwp card as:**

```
wwp:n  5  3  5  0  -1
```

- **Include the `mesh` card with cylindrical parameters.**
- **Run the problem.**

```
mcnp6  n=var4a  wwinp=var3be
```

- **Go to Exercise #4B**

Variance Reduction Exercise #3A

Use the weight window generator to create a weight window

- **Copy your input file from Exercise #2B.**
- **Add the weight window generator card:**

```
wwg  1  10
```

- **Run the problem.**

```
mcnp6  n = var3a
```

NOTE: if the input file is var3a, then the weight window output file will be var3ae, etc.

- **Complete the worksheet.**

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

Variance Reduction Exercise #3B

Use the weight window and generate a new one

- **Copy your input file from Exercise #3A.**
- **Add the `wwp` card as follows:**

```
wwp:n  5  3  5  0  -1
```

- **Keep the weight window generator card:**

```
wwg  1  10
```

- **Import your previous weight window file when running the problem:**

```
mcnp6  n=var3b  wwinp=var3ae
```

- **Complete the worksheet.**
- **Compare the two weight windows. What is the maximum percentage that any two corresponding cell values differ?**

Exercise #3B Selected Output

1neutron activity in each cell   print table 126

	cell	tracks entering	population	collisions	collisions * weight (per history)	number weighted energy	flux weighted energy	average track weight (relative)	average track mfp (cm)
1	10	120827	112833	865798	9.1432E+00	3.3524E-04	6.8756E-01	1.0440E+00	2.6583E+00
2	20	133059	100751	2843370	7.8855E+00	5.0136E-05	1.7672E-01	5.2273E-01	1.5844E+00
3	30	177504	87452	3528983	3.8604E+00	1.8487E-05	7.6292E-02	3.4832E-01	1.3409E+00
4	40	166220	88732	3263195	1.4670E+00	8.7550E-06	3.8518E-02	3.2428E-01	1.2387E+00
5	50	151414	91980	3120526	4.8364E-01	4.7461E-06	2.1516E-02	3.0218E-01	1.1876E+00
6	60	139235	92643	3012685	1.4805E-01	2.7680E-06	1.2640E-02	2.9331E-01	1.1607E+00
7	70	127152	93672	2851739	4.2690E-02	1.6561E-06	7.7650E-03	2.8477E-01	1.1441E+00
8	80	114161	99089	2830070	1.2143E-02	9.8790E-07	4.4446E-03	2.7614E-01	1.1329E+00
9	90	108127	103609	2814814	3.3069E-03	6.2081E-07	2.6579E-03	2.7533E-01	1.1270E+00
10	100	54977	76855	2070787	7.3674E-04	4.5322E-07	1.7961E-03	3.6597E-01	1.1249E+00
total		1292676	947616	27201967	2.3047E+01				

1status of the statistical checks used to form confidence intervals for the mean for each tally bin

```
 tally   result of statistical checks for the tfc bin (the first check not passed is listed) and error magnitude check for all bins

     1   passed the 10 statistical checks for the tally fluctuation chart bin result
         passed all bin error check:     1 tally bins all have relative errors less than 0.10 with no zero bins
```

Exercise #3B Selected Output

```
1tally fluctuation charts

                      tally     1
          nps         mean     error    vov  slope      fom
         8000   5.6510E-06 0.0858 0.0186   0.0     1037
        16000   6.0928E-06 0.0622 0.0183   4.0      978
        24000   6.0723E-06 0.0528 0.0155   5.2      895
        32000   6.1486E-06 0.0453 0.0104  10.0      898
        40000   6.1982E-06 0.0400 0.0077  10.0      910
        48000   6.2850E-06 0.0362 0.0062  10.0      919
        56000   6.2633E-06 0.0332 0.0051  10.0      937
        64000   6.2259E-06 0.0312 0.0043  10.0      936
        72000   6.2035E-06 0.0294 0.0043  10.0      930
        80000   6.1428E-06 0.0279 0.0038   8.0      937
        88000   6.1021E-06 0.0266 0.0035   7.9      936
        96000   6.1027E-06 0.0255 0.0033   7.7      932
       100000   6.0547E-06 0.0251 0.0031   8.1      932
```

Exercise #3B Selected Output

```
1neutron  weight-window lower bounds from the weight-window generator                                              print table 190

 energy:   1.000E+02
  cell
    10     5.000E-01
    20     1.006E-01
    30     4.634E-02
    40     1.751E-02
    50     5.794E-03
    60     1.786E-03
    70     5.187E-04
    80     1.470E-04
    90     4.092E-05
   100     1.123E-05
   110    -1.000E+00


1weight-window cards from the weight-window generator                                                               print table 200

          each card has ten leading blanks that must be removed by a text editor.

          wwp:n 5 3 5 0 0 0
          wwe:n  1.0000E+02
          wwn1:n    5.0000E-01  1.0058E-01  4.6337E-02  1.7511E-02  5.7938E-03
                    1.7861E-03  5.1867E-04  1.4704E-04  4.0925E-05  1.1229E-05
                   -1.0000E+00


1the following cells are bounded by cells with generated neutron
 weight-window bounds that are a factor of four or more different.


 neutron  window group  1     upper energy = 1.0000E+02

             window        maximum       window                 minimum       window
  cell       weight        neighbor      weight       ratio     neighbor      weight       ratio

    10     5.00000E-01          20     1.00582E-01     0.2          20     1.00582E-01     5.0
    20     1.00582E-01          10     5.00000E-01     5.0          30     4.63370E-02     2.2
```

Los Alamos NATIONAL LABORATORY EST.1943

UNCLASSIFIED

Exercise #3A & #3B Selected Output

Comparison of WWG Generated Weight Windows

Cell #	#3A	#3B	% Diff
1	5.0000E-01	5.0000E-01	0.00
2	1.0049E-01	1.0058E-01	0.09
3	4.6152E-02	4.6337E-02	0.40
4	1.7454E-02	1.7511E-02	0.33
5	5.7752E-03	5.7938E-03	0.32
6	1.7737E-03	1.7861E-03	0.70
7	5.2615E-04	5.1867E-04	1.44
8	1.5010E-04	1.4704E-04	2.08
9	4.0998E-05	4.0925E-05	0.18
10	1.1291E-05	1.1229E-05	0.55

NOTE: This agreement is considered unusually good.
It is because of the simple geometry.

Variance Reduction Exercise #4B

Generate weight windows on a cylindrical mesh while using mesh-based weight windows.

- **Copy your input file from Exercise #4A.**
- **Keep the `wwg` card as: `wwg 1 0`**
- **Keep the wwp card as: `wwp:n 5 3 5 0 -1`**
- **Keep the `mesh` card with cylindrical parameters.**
- **Run the problem.**

```
mcnp6 n=var4b wwinp=var4ae
```

- **Compare the two mesh-based weight window values. Do you iterate again or stop?**

Exercise #4B Selected Output

```
1status of the statistical checks used to form confidence intervals for the mean for each tally bin

 tally   result of statistical checks for the tfc bin (the first check not passed is listed) and error magnitude check for all bins

     1   passed the 10 statistical checks for the tally fluctuation chart bin result
         passed all bin error check:     1 tally bins all have relative errors less than 0.10 with no zero bins

 the 10 statistical checks are only for the tally fluctuation chart bin and do not apply to other tally bins.
```

```
1tally fluctuation charts

                        tally    1
      nps       mean      error   vov  slope    fom
     8000   5.1992E-06 0.0856 0.0196  0.0     619
    16000   6.2577E-06 0.0576 0.0114  5.7     623
    24000   6.6119E-06 0.0553 0.0868  3.5     439
    32000   6.5345E-06 0.0468 0.0574  3.4     463
    40000   6.2911E-06 0.0421 0.0431  3.7     466
    48000   6.3155E-06 0.0376 0.0325  4.3     483
    56000   6.1812E-06 0.0348 0.0268  4.2     488
    64000   6.1085E-06 0.0323 0.0223  4.6     498
    72000   6.1075E-06 0.0300 0.0188  4.8     510
    80000   6.1389E-06 0.0287 0.0171  4.1     501
    88000   6.1228E-06 0.0273 0.0147  4.5     506
    96000   6.0551E-06 0.0260 0.0132  4.7     513
   100000   6.0608E-06 0.0254 0.0123  5.1     515
```

Variance Reduction Exercise #5A

Generate & use weight windows on a rectangular mesh

- Copy your input file from Exercise #3B.
- Change the `wwg` card to: `wwg 1 0`
- Keep the wwp card as: `wwp:n 5 3 5 0 -1`
- Include the `mesh` card with rectangular parameters.
- Run the problem.

```
mcnp6  n=var5a  wwinp=var3be
```

- Go to Exercise #5B

Variance Reduction Exercise #5B

Generate weight windows on a rectangular mesh while using mesh-based weight windows.

- Copy your input file from Exercise #5A.
- Keep the `wwg` card as: `wwg 1 0`
- Keep the wwp card as: `wwp:n 5 3 5 0 -1`
- Keep the `mesh` card with rectangular parameters.
- Run the problem.

```
mcnp6  n=var5b  wwinp=var5ae
```

- Compare the two mesh-based weight window values. Do you iterate again or stop?

Exercise #5B Selected Output

```
1status of the statistical checks used to form confidence intervals for the mean for each tally bin


 tally    result of statistical checks for the tfc bin (the first check not passed is listed) and error magnitude check for all bins

     1    passed the 10 statistical checks for the tally fluctuation chart bin result
          passed all bin error check:     1 tally bins all have relative errors less than 0.10 with no zero bins


 the 10 statistical checks are only for the tally fluctuation chart bin and do not apply to other tally bins.

1tally fluctuation charts
```

```
                         tally     1
     nps          mean      error    vov  slope     fom
    8000   6.3336E-06 0.0875 0.0352  0.0     633
   16000   6.5981E-06 0.0590 0.0142  4.9     660
   24000   6.6558E-06 0.0494 0.0123  6.0     626
   32000   6.6659E-06 0.0441 0.0119  4.9     588
   40000   6.6688E-06 0.0398 0.0090  8.0     577
   48000   6.6123E-06 0.0369 0.0080  8.4     560
   56000   6.4465E-06 0.0339 0.0068  9.6     574
   64000   6.3980E-06 0.0316 0.0058 10.0     583
   72000   6.3783E-06 0.0297 0.0052  9.8     584
   80000   6.3693E-06 0.0282 0.0045 10.0     584
   88000   6.2415E-06 0.0271 0.0043  9.0     580
   96000   6.2915E-06 0.0258 0.0038 10.0     588
  100000   6.2839E-06 0.0254 0.0038 10.0     581
```

The "Final" Run

Now that you have surveyed the various techniques:

- **Select your variance reduction techniques and their "best" parameters.**
- **Make your final run. Increase your number of histories (nps) so that your final relative error is 1% or less.**
  - HINT: Use the expression for FOM to determine the time (histories) needed.
- **Compare your results (mean, R, VOV, FOM, slope) with those from your "selected" scoping run. How do they differ?**